A Method for Segmentation, Matching and Alignment of Dead Sea Scrolls*

Gil Levi, Pinhas Nisnevich, Adiel Ben-Shalom, Nachum Dershowitz, and Lior Wolf

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

The Dead Sea Scrolls are of great historical significance. Lamentably, in the decades since their discovery, many fragments have deteriorated. Fortunately, low-resolution grayscale infrared images of the Palestinian Archaeological Museum plates holding the scrolls in their discovered state are extant, along with recent high-quality multispectral images by the Israel Antiquities Authority. However, the necessary task of identifying each fragment in the new images on the old plates is tedious and time consuming to perform manually, and is often problematic when fragments have been moved from the original plate.

We describe an automated system that segments the new and old images of fragments from the background on which they were imaged, finds their matches on the old plates and aligns and superimposes them. To this end, we developed a deep-learning based segmentation method and a cascade approach for template matching, based on scale, shape analysis and dense matching.

We have tested the proposed method on five plates, comprising about 120 fragments. We present both quantitative and qualitative analyses of the results and perform an ablation study to evaluate the importance of each component of our system.

1. Introduction

The Dead Sea Scrolls, discovered in caves of the Judean desert during the years 1947–1956 and dating to the centuries around the turn of the eras, hold great historical, religious and linguistic significance. The tens of thousands of parchment and papyrus fragments include the oldest known manuscripts of many works later included in the Hebrew Bible canon, along with noncanonical and extra-biblical manuscripts in Hebrew, Aramaic and Greek that preserve important evidence of the diversity and richness of religious thought in late Second-Commonwealth Judea.

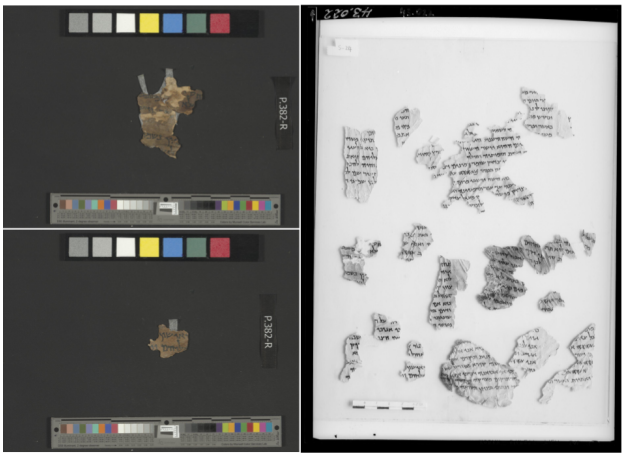

Figure 1. **Illustration of the matching problem**. Left: two samples of recent color images of fragments taken from plate 4Q57_382. Right: an old infrared image of plate M43.022. Our system successfully locates the two fragments on the left in the grayscale image of the whole plate. All images are provided courtesy of the Leon Levy Dead Sea Scrolls Digital Library, Israel Antiquities Authority; color photographer Shai Halevi, infrared by Najib Anton Albina.

Shortly after the scrolls were discovered, grayscale infrared images were taken by the Palestinian Archaeological Museum (PAM) of each plate on which they were stored, often containing dozens of small fragments. In the decades since their discovery, many plates have been reorganized. Currently, high-quality multispectral images are being taken at the Israel Antiquities Authority, and are offered to scholars and the public on the net (at http://www.deadseascrolls.org.il).

One of the tasks in this digitization project is that of locating each of the individual fragments on the old grayscale images and visually investigating how well they have been

*Research supported in part by Grant #01019841 from the Deutsch-Israelische Projektkooperation (DIP), Grant #1330/14 of the Israel Science Foundation (ISF), and a grant from the Blavatnik Family Fund. It forms part of P.N.'s M.Sc. research and G.L.'s Ph.D. research at Tel Aviv University. This research was made possible thanks to the Israel Antiquities Authority (IAA) Leon Levy Dead Sea Scrolls Digital Library's digital images. We would also like to thank Orit Rosengarten for her help.

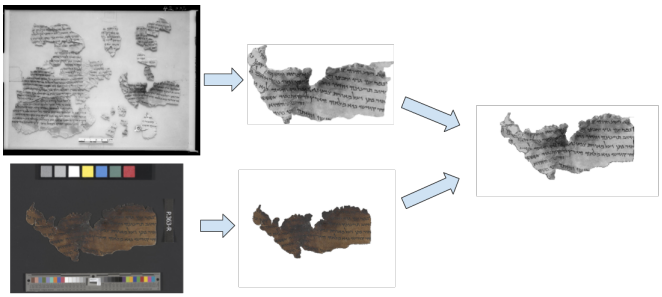

Figure 2. **System overview**. Our system begins by segmenting the old plate (top left) and the new color image (bottom left). Segmentation results of the old plate and the new fragment are shown at the top middle and bottom middle, respectively. Finally, after a match is found, the system aligns the two fragments (seen on the right).

preserved (see Figure 1). This is a wearisome chore since the fragments on a plate can look very similar and be extremely small – often no more than a square centimeter in size – and since many fragments are no longer on the same plate as when first imaged. Moreover, once the fragment is located, the conservator needs to visualize in her mind how the newer image is aligned with the old to reason how well it has been preserved. There is currently no automated tool to aid in this process.

We have developed a two-step method that both automatically locates a given fragment and aligns it against the old images. This will both save time for the conservation team and provide scholars with a composite image of the new segment overlayed on top of the old one.

First, we segment the fragments on the old plates to obtain a "pool" of candidates. This is challenging as the grayscale images of the old plates are of low resolution, the background has similar intensity to that of the scroll and there are shadows. We applied both an unsupervised technique based on edge detection [9] and a deep-learning technique that requires some manual labeling to train. Once we have a set of candidates, we employ a cascade approach to match the new high-quality color fragments to the corresponding low-quality grayscale candidate. The cascade is composed of a coarse scale test and a coarse shape test, followed by a more detailed dense matching [19, 14] test that compares the texture of the two fragments. Finally, we run dense matching again to align the matched fragments. The overall system is depicted in Figure 2.

We tested our method on five different plates containing a total of 120 fragments and present both quantitative and qualitative matching and alignment results. We further perform an ablation study to investigate the effectiveness of our system when removing some of the test in our matching cascade.

Our contributions include the following: (i) a deep-learning method for segmentation that is able to work under

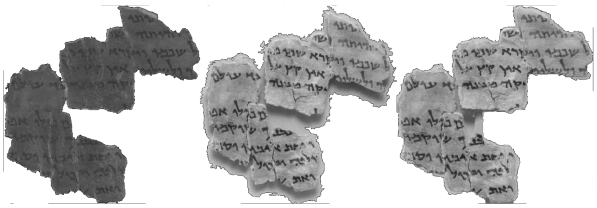

Figure 3. **Alignment example.** Left: new high-quality fragment image; middle: cropped fragment from old grayscale image; right: the fragment from the old image warped onto the recent image.

challenging conditions, which we compare with an edge-based method; (ii) a cascade approach for template matching, wherein each step in the cascade is responsible for a different visual feature (scale, shape and texture); and (iii) a dense-alignment method to help visualize any degradation a fragment suffered and to enable one to benefit from the combination of details visible in the old and new images.

2. Related work

Our method is related to two fundamental lines of work in vision: image segmentation and image registration.

Image segmentation. Image segmentation is one of the most studied and fundamental problems in computer vision. Its goal is to separate an image into multiple regions, where typically each region should correspond to some meaningful information, for example, objects or object parts [6, 40, 18, 20, 1, 29]. Early approaches to image segmentation include thresholding [26], edge detection [33], clustering [7, 28, 37], normalized cuts [35], active contours [5, 17, 39] and watershed [23, 34, 24].

Recently, deep convolutional neural networks (CNNs) which have proven successful in various computer vision applications [16, 12, 36], have also been applied to image segmentation. Specifically, fully convolutional neural networks (FCNNs) were found to be extremely useful for this task since they are competent in end-to-end training and feature generation and allow one to work with different input-image sizes. However, the drawback of using CNNs for segmentation is a decrease in segmentation resolution caused by pooling layers. This drawback hinders their applicability to the task at hand, as manuscript-fragment segmentation requires single-pixel level accuracy.

A number of encoder-decoder architectures have been suggested to cope with this problem, where the spatial information is gradually recovered by a decoder. These include: FCN [20], which performs semantic segmentation using end-to-end convolutional neural network and introduces skip connections that lead to better performance of deconvolution; SegNet [1], which transfers pooling indices to a decoder from an encoder that makes it more memory

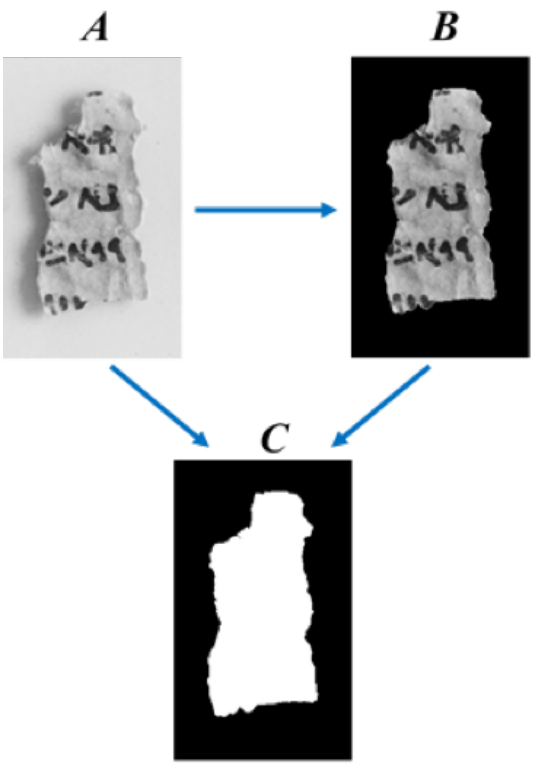

Figure 4. **Data preparation for training a deep segmentation network.** Input image *A* is transformed into the target image *C* via auxiliary manually-processed image *B*.

efficient; and u-net [29], which was shown to be a good solution for semantic segmentation relying on low amounts of data. Of the aforementioned architectures, u-net is the most applicable to a situation such as ours where the training set is limited, and we therefore based our network on this architecture.

Image registration. Image registration deals with finding local correspondences between images that depict the same content but have visual differences. Those differences can be due to illumination, motion blur, view point changes, image quality, etc. A thorough survey can be found in [41].

A common pipeline for image registration is composed of applying a local feature detector [31, 30, 13] to identify interest points in the image, extracting discriminative local features around those interest-points [21, 2, 4], matching them [3, 8, 11] and – if needed – aligning one image to another (e.g. by solving for an affine transformation between the images).

A different setting of the problem, one that can be viewed as a generalization of the local approaches, is optical flow, where the goal is to map each pixel in one image to its corresponding pixel in the other [15, 22]. Optical-flow methods lack the discriminative power of local features but enjoy the advantage of finding dense correspondence where the local methods are sparse in nature.

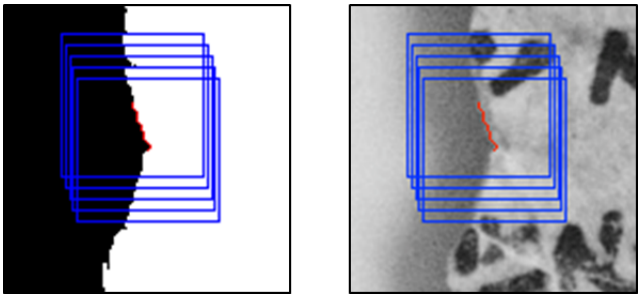

Figure 5. **Extracting training patches for segmentation.** By traversing the boundary of the target (left) and input (right) images, we extract training patches where each point on the boundary corresponds to the center of the patch. Blue squares represent patches and red curves show simultaneous traveling by boundaries of input and target images.

Recently, a marriage of the two, SIFT flow [19], has been proposed. In this method, SIFT features [21] are extracted at each pixel, and optical flow is computed between SIFT features instead of raw image intensities.

In addition to the general registration approaches described above, there are volumes of work specifically dedicated to document registration. Some approaches use line structure [10], template matching [27], projective geometry [32] and coarse-to-fine refinement [38].

The work most related to ours is probably [38], which applies a global alignment step followed by a local refinement step to handle misalignments. The method described herein is different in several respects: First, that work assumes the fragments have already been segmented, whereas we handle this task as well. Second, it assumes that a matching between a pair of fragments has already been performed, while we handle the matching task as well. Finally, we register the fragments using dense correspondence, which can express finer warping.

3. Method

In the next sections, we describe the pipeline that we have used to extract the fragments out of the plates and to separate them from the background. We first describe the segmentation algorithm that we used for the newly taken color images of the Dead Sea Scrolls, followed by a description of the pipeline for the old plates that were photographed during the 1950's.

3.1. New image segmentation

Recently, the Israel Antiquities Authority has begun a process of taking high resolution images of the all the Dead Sea Scroll fragments. They use a high resolution multi-wavelength imaging camera. Images of fragments are taken

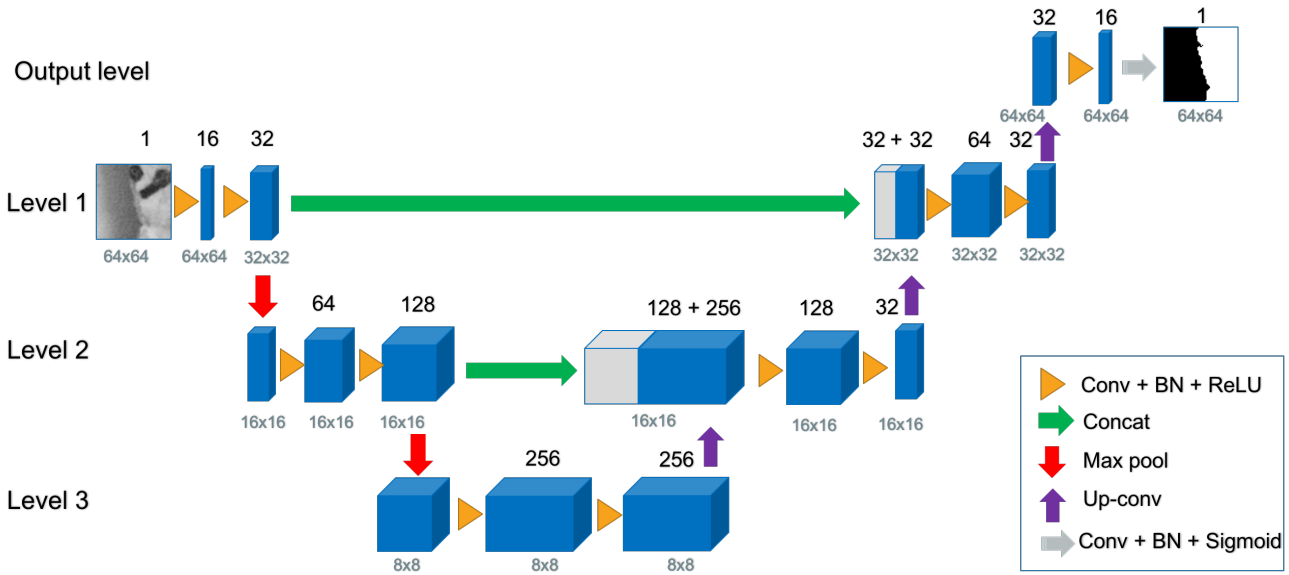

Figure 6. **Network architecture for Deep Segmentation.** Our network is based on an encoder-decoder u-net architecture [29] with three levels and an output level.

in different wavelengths in both infrared and visible light, and a color image is created as a composition of a subset of those. Each fragment is photographed on a black background together with the fragment label, a color target and ruler 1.

Since the fragments are placed on a dark felt surface, separating them from the background is not a simple task. The ink on the fragment closely resembles the dark background in color and the material preservation condition is generally poor, thus there are many holes in the fragment.

The first step in our segmentation pipeline is to extract connected components from the image and detect the fragment boundaries. We do this by converting the color image to grayscale and then setting a low threshold on the background intensity to extract the foreground components. This is followed by a *close* morphological operation and median filtering to remove noise and small holes.

Finally, we extract connected components from the binary image, disregard ones whose size is much smaller than the full image size and, from the remaining connected components, choose the one that is closest to the center of the binary image. This seemingly simple pipeline manages to segment the fragments well enough to perform matching.

3.2. Old plate segmentation

The older images of the plates that were photographed during the 50's require a different approach for fragment background separation. The images are grayscale and each plate can contain many small fragments. To segment them, we use an edge-detection segmentation method followed by connected components extraction. We use a state-of-the-art edge detection method as described in [9]. This method is based on a learned structured forests model [25] to detect edges.

This step is followed by setting a threshold on the grayscale edge image and computing connected components. We have found that extremely small connected components are usually noise and not real fragments; thus we discard connected components with area smaller than 0.1% of the whole plate. There are a few drawbacks with this method. First, if two fragment boundaries touch one another, it will not segment them accurately but will leave them connected together. A second drawback of this algorithm is that it is not able to handle shadows very well and often marks them as part of the fragment instead of marking them as background. Many of the fragments indeed do have shadows reflected on the plate because they were photographed at some small distance from the plate. For those fragments, the proposed edge-based segmentation method might detect shadow boundary as the true fragment boundary. For domain researchers whose goal is to re-connect fragments this poses a problem. However for the task of search and alignment it might be suffice.

Nevertheless, for more accurate results, we have developed a deep-network based segmentation algorithm. This algorithm is trained on manually segmented plates, and is able to remove shadow reflection more accurately. As opposed to our edge-based method, it is supervised, thus requiring training data that in our case must be manually labeled.

Figure 7. **Shape test.** On the left two column are segmented images of a candidate fragment pair. On the right are the two segmented fragments on top of one another. It can be visually seen that their Hamming distance corresponds to measuring how well the shapes overlap.

3.3. Network-based segmentation

Some of the scroll fragment images have a background that looks similar to the foreground near border points, and for this reason, correct labeling of each pixel is a difficult task even for humans who may mark some of border neighboring points incorrectly. To overcome these difficulties, a model based on a u-net architecture [29] was applied for binary segmentation of fragments.

Data preparation. For training, 21 grayscale images of scroll fragments were manually separated from the whitish background and from shadows, where they occur. For this purpose, initial photos (Figure 4*A*) were manually processed using a basic photo editor, where a background or a shadow were painted black, acquiring images of scroll fragments only (Figure 4*B*). Binary images of each of the grayscale images were created by labeling every pixel in the grayscale images by a binary label, where the foreground was labeled 1 and its background 0. Thusly, the background and the foreground were transformed into a black and white image (Figure 4*C*) by the following rule, using Iverson brackets: $C_{i,j} = [A_{i,j} = B_{i,j}]$, where $A_{i,j}$ and $B_{i,j}$ are the pixel values of images A and B, respectively, at the corresponding location (i, j) and $C_{i,j}$ is the resulting binary value of the binary image C at the same location (i, j), $i \in \{1, ..., w\}$, $j \in \{1, ..., h\}$, where w and h are the width and the height, respectively, of images A, B and C. (See Figure 4.)

By traversing the border of the segment in every binary image, 64×64 patches were extracted from both the original images *A* and their corresponding binary images *C*. Every boundary point served as the center of each patch. The patches were cropped from both *A* and *C* images by simultaneous following *C*'s boundary and corresponding location on image *A* – as presented in Figure 5. Figure 4*C* shows the target image *C* for the network.

Network architecture. Our network is an encoder/decoder type of architecture based on the u-net model of fully convolutional network suggested in [29]. In this model, the spatial dimension of the image is gradually reduced by the encoder with max-pooling layers, whereas the decoder retrieves spatial dimension and details of the image. The model is suitable for a case of a limited number of training images, nevertheless achieving precise segmentations.

Our network architecture is composed of two paths: a contracting path (left) and an expansive one (right), which can be viewed as four levels l_1, l_2, l_3, plus output, as illustrated in Figure 6. Each level l_i, $i = 1, 2, 3$, consists of two 3×3 convolution layers and one 2×2 max pooling layer with a stride of 2 pixels on levels l_1 and l_2, where each convolution was followed by batch normalization (BN) and rectified linear unit (ReLU) layers.

A concatenation of the feature maps from each level l_{i-1} and l_i was performed by upsampling the feature map at level l_i with a 3×3 convolution. After concatenation, two 3×3 convolutions were applied, each convolution followed by batch normalization and ReLU layers. At the output level the final activation function was sigmoid instead of ReLU.

Network training and the loss used. As described above, the network was trained using (X, Y) as a training data, where X is a set of 64×64 grayscale images and Y is a set of 64×64 binary images. For every $i \in \{1, \ldots, m\}$ $Y^{(i)}$ is the target image for $X^{(i)}$, where m is the size of the training data. ADADELTA was used as an optimization method with learning rate of 0.01 and a batch size of 1200. The loss function was defined as follows:

$$loss = \frac{1}{5n} \sum_{i,j} \|Y_{i,j} - \hat{Y}_{i,j}\|_1 + \|\hat{Y}_{i-1,j} - \hat{Y}_{i,j}\|_1 + \|\hat{Y}_{i,j-1} - \hat{Y}_{i,j}\|_1 + \|\hat{Y}_{i-1,j-1} - \hat{Y}_{i,j}\|_1 + \|\hat{Y}_{i-1,j+1} - \hat{Y}_{i,j}\|_1.$$

where n is the batch size and $\hat{Y}$ is the network prediction. This loss function encourages both adherence to the ground truth as well as additional smoothness of the predicted targets $\hat{Y}$.

Segment extraction. A trained network with a threshold θ was used on a plate image I providing segments prediction image $\bar{Y} := [\hat{Y}(I) > \theta] \cdot I$. Each segment was then extracted using a connected-components algorithm.

3.4. Template matching and alignment

After applying segmentation, we obtain a pool of *new fragments* (taken from the recent high-quality color images) S_i, $i = 1, \ldots, n$, that need to be matched against a pool of *old fragments* (taken from the old grayscale images) $P_j, j = 1, \ldots, m$. Next, we employ a cascade approach, in which each step handles a particular visual feature of a candidate pair (S_{i_0}, P_{j_0}).

Plates	Matches	Edge Segmentation				Deep Segmentation			
		#TP	#FP	Recall	FPR	#TP	#FP	Recall	FPR
4Q57_363, M42.002	5	3	0	60%	0.0%	3	1	60%	1.3%
4Q57_382, M42.000	8	5	17	63%	5.3%	8	24	100%	7.6%
4Q57_382, M43.022	13	5	8	38%	2.7%	8	9	62%	3.0%
4Q57_382, M43.162	3	1	15	33%	2.6%	1	13	33%	2.2%
4Q57_387, M43.029	18	8	76	44%	4.5%	5	91	27%	5.4%

Table 1. **Matching results.** For each set of recent fragment images and corresponding grayscale plate, we show the number of correct matches found (true positives) – out of the total number of fragments on the plate, as well as the number of false matches (false positives), the true positive rate (recall) and false positive rate (FPR).

Scale test. First, we apply a simple scale test to filter out candidate pairs in which one fragment is much larger than the other. The new images S_i were taken at an aspect ratio that is about 3 times larger than that of the old images. Assume S_{i_0} is $k \times l$ pixels and P_{j_0} is $t \times p$ pixels. If $|k - 3t| > T_{scale}$ or $|l - 3p| > T_{scale}$, we discard the pair. We allow a T_{scale} that is large enough to serve as a coarse test and will never filter out correct matches.

Shape test. Now that we have filtered out candidate pairs that are vastly different in scale, the second appearance feature we filter for is shape. We found the following simple test to be extremely useful: resize S_{i_0} to the size of P_{j_0}, viz. $t \times p$, and take both segments' binary masks by thresholding them (since we applied segmentation, the background is constant and a simple threshold operation is enough). Denote by S'_{i_0} and P'_{j_0} the binary masks, we compute their Hamming distance $\|S'_{i_0} - P'_{j_0}\| = \sum_{\alpha=1}^{t} \sum_{\beta=1}^{p} (s'_{\alpha,\beta} \textbf{ xor } p'_{\alpha,\beta})/(tp)$ and discard the candidate pair if this value is above a predefined threshold T_{shape}. Again, T_{shape} is set such that this test will filter out fragments that are markedly different, but will be coarse enough not to eliminate any correct matches. This process is illustrated in Figure 7.

Dense matching. The final step in our cascade is based on dense matching and compares the texture and content of the two images at a finer resolution. We leverage the SIFT flow algorithm [19]. SIFT flow begins by extracting SIFT [21] descriptors at every pixel of the fragments. Next, the algorithm aims to find the flow $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$, that is, the displacement of the SIFT feature at pixel $\mathbf{p} = (x, y)$ at the first image to its location $(x + u(\mathbf{p}), y + v(\mathbf{p}))$ at the second image. This is done by solving the following optimization problem: Let s_1 and s_2 be the two SIFT images to match. Set ε contains all the spatial neighborhoods (a four-neighbor system is used). The energy function for

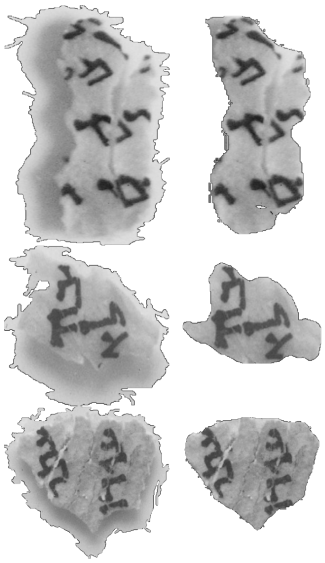

Figure 8. **Segmentation results**. We compare between edge segmentation (left) and deep segmentation (right). Notice that shadows remain in edge segmentation, while deep segmentation successfully eliminates them.

SIFT flow is defined as:

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \min\left(\|s_1(\mathbf{p}) - s_2(\mathbf{p} + \mathbf{w}(\mathbf{p}))\|_1, t\right) + \quad (1)$$

$$\sum_{\mathbf{p}} \eta\left(|u(\mathbf{p})| + |v(\mathbf{p})|\right) + \quad (2)$$

$$\sum_{(\mathbf{p},\mathbf{q}) \in \varepsilon} \min\left(\alpha|u(\mathbf{p}) + u(\mathbf{q})|, d\right) + \min\left(\alpha|v(\mathbf{p}) + v(\mathbf{q})|, d\right) \quad (3)$$

Term (1) constrains the corresponding SIFT features to be similar, term (2) regularizes the flow vectors so that they are small (η is a parameter) and term (3) forces adjacent pixels to have similar flow values. Optimization is done using dual-layer loopy belief propagation.

To use SIFT flow as a matching test, we resize the pair of fragments to be the same, apply SIFT flow and use the minimal value of the energy function as a matching score.

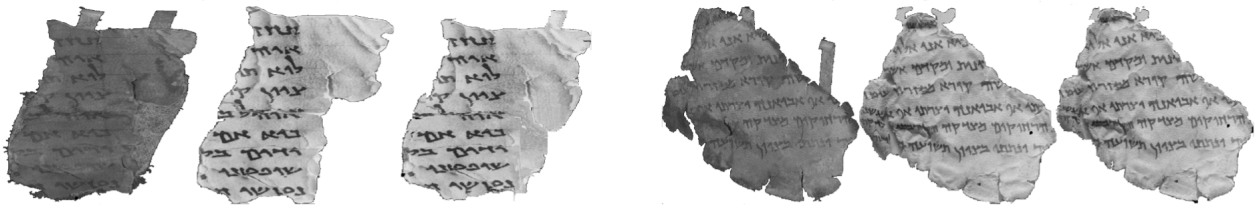

Figure 9. **Qualitative alignment results.** From left to right: newly taken image of a fragment from set 4Q57_382, corresponding fragment from plate M43.022, registration results of the two; newly taken image of a fragment from set 4Q57_382, corresponding fragment from plate M42.000, registration of the two.

If the score is below a predetermined threshold T_{flow}, then the candidate pair S_{i_0}, P_{j_0} is considered a match.

Alignment. Finally, to align the two matched fragments S_{i_0}, P_{j_0}, we resize P_{j_0} to the dimensions of S_{i_0} (the new images S_i were taken at higher resolution), run SIFT flow again and warp P_{j_0} using the flow $\mathbf{w_0}$ obtained. The alignment is illustrated in Figure 3.

4. Experiments

We apply our proposed pipeline to sets 4Q57_363, 4Q57_382 and 4Q57_387 of the recent high-quality color images, consisting of fragments of Isaiah found in Cave 4 at Qumran written on parchment and dating to the Herodian period. Roughly, 4Q57_363 corresponds to PAM plate M42.002, 4Q57_382 corresponds to M42.000, M43.022 and M43.162 and 4Q57_387 corresponds to M43.029.

There are five fragments in 4Q57_363, all of which appear on plate M42.002, along with 12 other fragments in M42.002 that are not related to 4Q57_363. There are 18 different fragments in 4Q57_382, 8 of which appear on M42.000, 13 on M43.022 and 3 on M43.162. Some of the fragments appear on more than one PAM plate. Also, M42.000, M43.022 and M43.162 contain, respectively, 10, 4 and 29 fragments that are unrelated to 4Q57_382. Finally, 4Q57_387 contains 54 fragments; 18 appear on M43.029 along with 13 unrelated fragments. In our experiments, we set T_{scale} to be 400 pixels, T_{shape} to be 0.3 and T_{flow} to be 0.7.

4.1. Quantitative results

In each of our experiments, we chose a pair of recent image set and its corresponding PAM plate, applied both of our segmentation methods to the fragments and finally ran our matching and alignment cascade on all candidate pairs of recent and old fragments. This amounts to a large number of candidate pairs: when matching 4Q57_363 to M42.002, there is a total of 85 candidate pairs, but only 5 of them are positive matches; when matching 4Q57_382 to M42.000 there is a total of 324 candidate pairs, only 8 of which are positives; 4Q57_382 to M43.022 – 306 candidate pairs, 13 are positives; 4Q57_382 to M43.162 – 576 candidate pairs, 3 are positives and finally when matching 4Q57_387 to M43.029 there is a total of 1674 candidate pairs to consider, only 18 of which are positives.

We summarize the quantitative results in Table 1. Due to the imbalance between the amount of negative and positive samples in our test set, we report matching results in terms of both the number of correct matches found (true positives) and the number false matches reported (false positives), and also give the recall and false positive rates.

As can be seen in Table 1, our method is able to achieve markedly high recall at low false-positive rates. Notice the improved performance of the deep segmentation method in almost all cases.

4.2. Ablation study

To further investigate the effect of the different components in our system, we perform an ablation study. To this end, we repeat the experiments described in Section 4.1 for plates 4Q57_363-M42.002 and 4Q57_382-M42.000, each time removing certain tests in our matching cascade. The results are presented in Table 2. Note that each of the tests in the cascade contributes to eliminating false matches, where the most important test is measuring dense-correspondence using SIFT flow, following by the scale test. Also note that removing any of the tests does not increase the number of correct matches.

4.3. Comparison of segmentation methods

We compared the results obtained by the two methods – edge segmentation and deep segmentation. The first method enabled the system to extract scroll fragments from a plate image but failed to distinguish between shadows, background and scroll edge sections (Fig. 8, left), whereas with the second method it was possible to isolate scroll fragments containing written text, thus acquiring the scroll segment only (Fig. 8, right). It is feasible to find and align fragments obtained by edge segmentation, but the combination of fragments obtained by edge segmentation cannot lead to continuous text lines since shadows and edge sections interfere with fragment coupling.

Figure 10. **Failure cases.** The two leftmost figures present false matches of plates 4Q57_382-M43.022 and 4Q57_382-M43.162, respectively. The rightmost figure presents a correct match that resulted in a poor registration result. Analyzing these and several other examples showed that such problems are partially due to errors in the color image segmentation and that, generally, mismatches are more common when matching very small segments, as one would expect.

Plates	Matches	Scale	Shape	Flow	Edge Segmentation				Deep Segmentation			
					#TP	#FP	Recall	FPR	#TP	#FP	Recall	FPR
4Q57_363, M42.002	5	✓	✓	✗	3	4	60%	5.0%	3	3	60%	3.7%
4Q57_382, M42.000	8	✓	✓	✗	5	74	63%	23.4%	8	101	100%	31.9%
4Q57_363, M42.002	5	✓	✗	✓	3	1	60%	1.3%	3	2	60%	2.5%
4Q57_382, M42.000	8	✓	✗	✓	5	38	63%	12.0%	8	72	100%	22.7%
4Q57_363, M42.002	5	✗	✓	✓	3	1	60%	1.3%	3	3	60%	3.7%
4Q57_382, M42.000	8	✗	✓	✓	5	22	63%	6.9%	8	35	100%	11.0%
4Q57_363, M42.002	5	✓	✗	✗	3	21	60%	26.3%	3	15	60%	18.8%
4Q57_382, M42.000	8	✓	✗	✗	6	139	75%	43.9%	8	234	100%	74.0%
4Q57_363, M42.002	5	✗	✓	✗	3	16	60%	20.0%	3	20	60%	25.0%
4Q57_382, M42.000	8	✗	✓	✗	5	114	63%	36.0%	8	186	100%	58.8%
4Q57_363, M42.002	5	✗	✗	✓	3	3	60%	3.8%	3	6	60%	7.5%
4Q57_382, M42.000	8	✗	✗	✓	5	74	60%	23.4%	8	131	100%	41.4%

Table 2. **Ablation results.** We present matching performance when removing various tests from our matching cascade.

4.4. Qualitative registration results

In addition to numerical results for matching, we also present registration results in Figure 9. As can be seen, the ink appears faded on the newly taken images while appearing much sharper on the early grayscale images. Using such visualizations, scholars and conservators can deduce the level of deterioration of the scrolls and their preservation conditions.

4.5. Failure examples

To help analyze the performance of the proposed method, we give examples of false matches and of a correct match with poor registration in Figure 10. Such errors mostly occur when matching very small fragments.

5. Conclusions

Faced with a real-world demand for a robust solution that can align multiple fragments, we have designed a practical system that incorporates tools from multiple areas of computer-vision research.

For fragment segmentation, we proposed an edge-based segmentation method that does not require training, as well as a second deep segmentation method for improved performance, requiring some initial manual labeling to train. For matching and aligning segmented fragments, we designed a cascade method that is able to obtain high accuracy while maintaining a low false positive rate and is able to handle fragments that are extremely difficult to locate (see Figure 1).

The feedback we have received from scholars has been extremely positive and encouraging. Our system alleviates much of the difficulty in fragment matching and registration and will dramatically save time for archaeologists investigating the scrolls. Superimposing multiple images of fragments provides scholars with a new tool for determining how well they have been preserved. Furthermore, our segmentation method can be used for other preservation efforts as well, for example connecting different fragments of the same original scroll.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 3

[3] M. Brown and D. G. Lowe. Recognising panoramas. In *Proceedings ICCV*, volume 3, page 1218, 2003. 3

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision – ECCV 2010*, pages 778–792, 2010. 3

[5] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 2

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv preprint arXiv:1606.00915*, 2016. 2

[7] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1):9–15, 2006. 2

[8] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2029, 2015. 3

[9] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *ArXiv*, 2014. 2, 4

[10] K.-C. Fan, Y.-K. Wang, and M.-L. Chang. Form document identification using line structure based features. In *Document Analysis and Recognition, 2001. Proceedings Sixth International Conference on*, pages 704–708. IEEE, 2001. 3

[11] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21, 2015. 3

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 2

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, pages 147–152. Manchester, UK, 1988. 3

[14] T. Hassner and C. Liu, editors. *Dense Image Correspondences for Computer Vision*. Springer, 2016. 2

[15] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, 1981. 3

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2

[17] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 316–323. IEEE, 2000. 2

[18] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*, 2016. 2

[19] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 2, 3, 6

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3, 6

[22] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, volume 2, pages 674–679, Vancouver, BC, Aug. 1981. 3

[23] A. P. Mangan and R. T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, 1999. 2

[24] H. P. Ng, S. H. Ong, K. W. C. Foong, P. S. Goh, and W. L. Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006. 2

[25] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4):185–365, 2011. 4

[26] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 2

[27] H. Peng, F. Long, and Z. Chi. Document image recognition based on template matching of component block projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1188–1192, 2003. 3

[28] S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pages 137–143. Calcutta, India, 1999. 2

[29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2, 3, 4, 5

[30] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Computer Vision – ECCV 2006*, pages 430–443, 2006. 3

[31] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119, 2010. 3

[32] R. Safari, N. Narasimhamurthi, M. Shridhar, and M. Ahmadi. Document registration using projective geometry. *IEEE Transactions on Image Processing*, 6(9):1337–1341, 1997. 3

[33] N. Senthilkumaran and R. Rajesh. Edge detection techniques for image segmentation–a survey of soft computing approaches. *International Journal of Recent Trends in Engineering*, 1(2):250–254, 2009. 2

[34] L. Shafarenko, M. Petrou, and J. Kittler. Automatic watershed segmentation of randomly textured color images. *IEEE Transactions on Image Processing*, 6(11):1530–1544, 1997. 2

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 2

[36] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pages 1701–1708, 2014. 2

[37] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993. 2

[38] Y. Zarai, T. Lavee, N. Dershowitz, and L. Wolf. Integrating copies obtained from old and new preservation efforts. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 47–51. IEEE, 2013. 3

[39] K. Zhang, L. Zhang, H. Song, and W. Zhou. Active contours with selective local or global segmentation: a new formulation and level set method. *Image and Vision Computing*, 28(4):668–676, 2010. 2

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 2

[41] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. 3